

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»  
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ  
КАФЕДРА «ІНФОРМАЦІЙНИХ СИСТЕМ»

Лабораторна робота № 12

з дисципліни «Операційні системи»

Тема: *«Програмування міжпроцесної та багатопоточної взаємодії»*

**Виконав:**

Студент групи AI-202

Матненко Станіслав Володимирович

Одеса 2020

**Мета роботи:** вивчити особливості обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізацію процесів через семафори та м'ютекси.

### **Завдання до роботи:**

#### **2.1 Робота з іменованими каналами**

2.1.1 В домашньому каталозі вашого користувача створіть іменований канал з використанням команди `mkfifo`:

- назва каналу співпадає з вашим прізвищем у транслітерації
- права доступу до каналу ( можна лише читати та писати власнику).

2.1.2 Підключіть до іменованого каналу процес, який буде в нього писати за такими командами:

- отримати зміст каталогу `/etc`
- отримати назви файлів, які починаються з букви вашого прізвища у транслітерації.

2.1.3 Перейдіть до нового терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу.

2.1.4 Поверніться до 1-го терміналу та підключіть до іменованого каналу процес, який буде в нього писати, архівуючи файл командою `gzip -c < pipe > file1.gz` де `pipe` – назва вашого каналу, `file1.gz` – назва файлу, який буде створено в результаті архівації

2.1.5 Перейдіть до 2-го терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу, архівуючи файл `/etc/passwd`

#### **2.2 Програмування іменованих каналів**

Повторіть попереднє завдання, але пункт 2.1.1 виконайте через програмування іменованого каналу за прикладом з рисунку 1.

#### **2.3 Програмування потоків**

За прикладом з рисунку 2 розробіть програму керування потоками, в якій в повідомленнях буде вказано ваше прізвище латиницею.

Виконайте програму за вказаним прикладом.

#### **2.4 Програмування семафорів**

За прикладом з рисунку 3 розробіть програму керування семафором, в якій в повідомленнях буде вказано ваше прізвище латиницею. Виконайте програму в двох терміналах за вказаним прикладом.

## Хід роботи

### 2.1 Робота з іменованими каналами

2.1.1 В домашньому каталозі вашого користувача створіть іменований канал з використанням команди `mkfifo`:

- назва каналу співпадає з вашим прізвищем у транслітерації
- права доступу до каналу ( можна лише читати та писати власнику).

```
[matnenko_stanislaw@vpsj3IeQ ~]$ mkfifo matnenko
[matnenko_stanislaw@vpsj3IeQ ~]$ chmod u=rw matnenko
```

2.1.2 Підключіть до іменованого каналу процес, який буде в нього писати за такими командами:

- отримати зміст каталогу `/etc`
- отримати назви файлів, які починаються з букви вашого прізвища у транслітерації.

```
[matnenko_stanislaw@vpsj3IeQ ~]$ ls /etc/m* > matnenko
```

2.1.3 Перейдіть до нового терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу.

```
[matnenko_stanislaw@vpsj3IeQ ~]$ cat matnenko
/etc/machine-id
/etc/magic
/etc/mail.rc
/etc/makedumpfile.conf.sample
/etc/man_db.conf
/etc/mke2fs.conf
/etc/motd
/etc/mtab
/etc/my.cnf
```

2.1.4 Поверніться до 1-го терміналу та підключіть до іменованого каналу процес, який буде в нього писати, архівуючи файл командою `gzip -c < pipe > file1.gz` де `pipe` – назва вашого каналу, `file1.gz` – назва файлу, який буде створено в результаті архівації

```
[matnenko_stanislaw@vpsj3IeQ ~]$ gzip -c <matnenko> file_matnenko.gz
```

2.1.5 Перейдіть до 2-го терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу, архівуючи файл `/etc/passwd`

```
[matnenko_stanislaw@vpsj3IeQ ~]$ cat /etc/passwd > matnenko
```

Одержаний файл:

```
file_matnenko.gz 2573 May 25 20:10
```

## 2.2 Програмування іменованих каналів

Повторіть попереднє завдання, але пункт 2.1.1 виконайте через програмування іменованого каналу за прикладом з рисунку 1.

Скрипт:

```
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>
#include <stdio.h>

#define NAMEDPIPE_NAME "matnenko"
#define BUFSIZE 50

int main (int argc, char ** argv) {
    int fd, len;
    char buf[BUFSIZE];

    if ( mkfifo(NAMEDPIPE_NAME, 0777) ) {
        fprintf(stderr, "Error in mkfifo!");
        return 1;
    }
    printf("%s is created\n", NAMEDPIPE_NAME);

    if ( (fd = open(NAMEDPIPE_NAME, O_RDONLY)) <= 0 ) {
        fprintf(stderr, "Error in open!");
        return 1;
    }
    printf("%s is opened\n", NAMEDPIPE_NAME);

    do {
        memset(buf, '\0', BUFSIZE);
        if ( (len = read(fd, buf, BUFSIZE-1)) <= 0 ) {
            printf("END!");
            close(fd);
            remove(NAMEDPIPE_NAME);
            return 0;
        }
        printf("Incomming message (%d): %s\n", len, buf);
    } while ( 1 );
}
```

Результат виконання:

1-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ ./fifo
matnenko is created
matnenko is opened
Incomming message (49): adjtime
aliases
aliases.db
alternatives
anacronta
Incomming message (49): b
asound.conf
audisp
audit
bash_completion.d
bash
Incomming message (49): rc
binfmt.d
-----
```

1-ий термінал

```
END! [matnenko_stanislaw@vpsj3IeQ ~]$ ./fifo
matnenko is created
matnenko is opened
Incomming message (49): /etc/machine-id
/etc/magic
/etc/mail.rc
/etc/make
Incomming message (49): dumpfile.conf.sample
/etc/man_db.conf
/etc/mke2fs
Incomming message (49): .conf
/etc/motd
/etc/mtab
/etc/my.cnf
```

2-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ ls /etc > matnenko
```

2-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ ls /etc/m* > matnenko
```

## 1-ий термінал

```
|END! [matnenko_stanislaw@vpsj3IeQ ~]$ gzip -c < ./fifo > file matnenkol.gz
```

## 2-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ cat /etc/passwd > matnenko  
[matnenko_stanislaw@vpsj3IeQ ~]$
```

Одержаний файл:

```
file_mat~nkol.gz 2859 May 25 20:18
```

## 2.3 Програмування потоків

За прикладом з рисунку 2 розробіть програму керування потоками, в якій в повідомленнях буде вказано ваше прізвище латиницею.

Виконайте програму за вказаним прикладом.

Скрипт:

```
#include <stdio.h>
#include <pthread.h>

main() {
    pthread_t f2_thread, f1_thread;
    void *f2(), *f1();
    int i1 = 10, i2 = 10;
    pthread_create(&f1_thread, NULL, f1, &i1);
    pthread_create(&f2_thread, NULL, f2, &i2);
    pthread_join(f1_thread, NULL);
    pthread_join(f2_thread, NULL);
}

void *f1(int *x) {
    int i, n;
    n = *x;
    for (i=1; i<n; i++) {
        printf("f1: Matnenko\n");
        sleep(1);
    }
    pthread_exit(0);
}

void *f2(int *x) {
    int i, n;
    n = *x;
    for (i=1; i<n; i++) {
        printf("f2: Matnenko\n");
        sleep(1);
    }
    pthread_exit(0);
}
```

Результат виконання:

```
[matnenko_stanislaw@vpsj3IeQ ~]$ nano thread.c
[matnenko_stanislaw@vpsj3IeQ ~]$ gcc thread.c -o thread -lpthread
[matnenko_stanislaw@vpsj3IeQ ~]$ ./thread
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
f1: Matnenko
f2: Matnenko
```

## 2.4 Програмування семафорів

За прикладом з рисунку 3 розробіть програму керування семафором, в якій в повідомленнях буде вказано ваше прізвище латиницею. Виконайте програму в двох терміналах за вказаним прикладом.

Скрипт:

```
#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>
#include <stdio.h>

#define SEMAPHORE_NAME "/matnenko_semaphore"

int main(int argc, char ** argv) {

    sem_t *sem;

    if ( argc != 2 ) {
        if ((sem = sem_open(SEMAPHORE_NAME, O_CREAT, 0777, 0)) == SEM_FAILED ) {
            fprintf(stderr, "sem_open error");
            return 1;
        }
        printf("sem_open. Semaphore Matnenko is taken.\nWaiting for it to be dropped.\n");
        if (sem_wait(sem) < 0 )
            fprintf(stderr, "sem_wait error");
        if ( sem_close(sem) < 0 )
            fprintf(stderr, "sem_close error");
        return 0;
    }
    else {
        printf("Dropping semaphore Matnenko...\n");
        if ( (sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED ) {
            fprintf(stderr, "sem_open error");
            return 1;
        }
        sem_post(sem);
        printf("sem_post. Semaphore Matnenko dropped.\n");
        return 0;
    }
}
```

Результат виконання:

### 1-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ nano semaphore.c
[matnenko_stanislaw@vpsj3IeQ ~]$ gcc semaphore.c -o semap -lpthread
[matnenko_stanislaw@vpsj3IeQ ~]$ ./semap
sem_open. Semaphore Matnenko is taken.
Waiting for it to be dropped.
```

### 2-ий термінал

```
[matnenko_stanislaw@vpsj3IeQ ~]$ ./semap 1
Dropping semaphore Matnenko...
sem_post. Semaphore Matnenko dropped.
```